

UNIVERSIDAD DON BOSCO  
FACULTAD DE INGENIERIA  
ESCUELA DE COMPUTACION



MATERIA:

SISTEMAS EXPERTOS

CATEDRATICO:

ING. CRUZ ANTONIO GALDAMEZ RIVERA

ALUMNOS:

WILLIAM ERNESTO COCA FLORES CF990179

GILBERTO ANTONIO VELÁSQUEZ GARCÍA VG970225

GRUPO

01

SOYAPANGO, 30 DE JUNIO 2004

### *La Definición de un Sistema Experto:*

**Sistema experto**, tipo de programa de aplicación informática que adopta decisiones o resuelve problemas de un determinado campo, como las finanzas o la medicina, utilizando los conocimientos y las reglas analíticas definidas por los expertos en dicho campo. Los expertos solucionan los problemas utilizando una combinación de conocimientos basados en hechos y en su capacidad de razonamiento. En los sistemas expertos, estos dos elementos básicos están contenidos en dos componentes separados, aunque relacionados: una base de conocimientos y una máquina de deducción, o de inferencia. La base de conocimientos proporciona hechos objetivos y reglas sobre el tema, mientras que la máquina de deducción proporciona la capacidad de razonamiento que permite al sistema experto extraer conclusiones. Los sistemas expertos facilitan también herramientas adicionales en forma de interfaces de usuario y los mecanismos de explicación. Las interfaces de usuario, al igual que en cualquier otra aplicación, permiten al usuario formular consultas, proporcionar información e interactuar de otras formas con el sistema. Los mecanismos de explicación, la parte más fascinante de los sistemas expertos, permiten a los sistemas explicar o justificar sus conclusiones, y también posibilitan a los programadores verificar el funcionamiento de los propios sistemas.

### *Marco Teorico*

Estas pensando que la inteligencia artificial es una cosa de los últimos 3, 5 o como mucho los 10 últimos años, pero NO!, los primeros pasos en la inteligencia artificial se dieron en los !AÑOS 50!.

A comienzos de los años 50 el conocido Alan Mathinsong Turing publicó "Inteligencia y Funcionamiento de las Máquinas" con el fin de demostrar hasta que punto estas tienen inteligencia.

En estos años se dieron varias definiciones de lo que significaba la inteligencia en una máquina. Sobre lo que denominamos la inteligencia artificial.

Uno de los primeros sistemas expertos se llamo Dendral y era capaz de calcular o descubrir hechos relativos a la estructura molecular a partir de unos datos químicos sin elaborar. Otro sistemas expertos famosos son MYCIN que diagnostica enfermedades de la sangre y que sugiere un tratamiento y PUFF, un sistema similar pero para enfermedades de pulmón.

En el año 1950 el campo de la automática recibe un gran impulso cuando Wiener desarrolla el principio de la retroalimentación. La teoría de la retroalimentación es base fundamental de los sistemas de control.

En 1955 Newell y Simon desarrollan la Teoría de la lógica. Este desarrollo permitió desarrollar un programa que exploraba la solución a un problema utilizando ramas y nudos, seleccionando únicamente las ramas que más parecían acercarse a la solución correcta del problema.

En 1956, se celebra una conferencia en Vermont (USA) de gran trascendencia en el desarrollo de la I.A. John McCarthy propone por primera vez el uso del término "Inteligencia Artificial" para denominar el estudio del tema.

En 1957, aparece la primera versión de "The General Problem Solver" (GPS, Solucionador general de problemas), un programa capaz de solucionar problemas de sentido común pero no problemas del mundo real como diagnósticos médicos. El GPS utilizaba la teoría de la retroalimentación de Wiener..

En 1958 McCarthy anuncia su nuevo desarrollo el lenguaje LISP (LISt Procesing), el lenguaje de elección para todos aquellos desarrolladores inmersos en el estudio de la IA.

En 1963, el Instituto Tecnológico de Massachussets (MIT) recibe una subvención de 2,2 millones de dólares del gobierno de los Estados Unidos en concepto de investigación en el campo de la IA. De esa forma, se comprueba la importancia que el Gobierno concede a la investigación dentro de ese campo.

En 1965 aparece DENDRAL, el primer sistema experto. Es en ese año cuando Feigenbaum entra a formar parte del departamento de informática de Stanford. Allí conoció a Joshua Lederberg, el cual quería averiguar cual era la estructura de las moléculas orgánicas

completas. El objetivo de DENDRAL fue estudiar un compuesto químico. El descubrimiento de la estructura global de un compuesto exigía buscar en un árbol las posibilidades, y por esta razón su nombre es DENDRAL que significa en griego "árbol". Antes de DENDRAL los químicos solo tenían una forma de resolver el problema, esta era tomar unas hipótesis relevantes como soluciones posibles, y someterlas a prueba comparándolas con los datos.

La realización de DENDRAL duró más de diez años (1965-1975). Se le puede considerar el primer sistema experto.

En 1965 también se empezaron a utilizar técnicas para la resolución de problemas que se caracterizaban por la búsqueda heurística como modelo para la resolución de problemas, y con ellas comenzó la investigación y desarrollo de los sistemas expertos.

En 1972, en la Universidad de Standford se desarrolla MYCIN, sistema experto dentro del campo de la medicina para diagnóstico de enfermedades infecciosas en la sangre. MYCIN se trataba de un sistema experto para el diagnóstico de enfermedades infecciosas. Desde los resultados de análisis de sangre, cultivos bacterianos y demás datos, el programa era capaz de determinar, o en lo menos, sugerir el microorganismo que estaba causando la infección. Después de llegar a una conclusión, MYCIN prescribía una medicación que se adaptaba perfectamente a las características de la persona, tales como el peso corporal de este.

Al mismo tiempo, David Marr propone nuevas teorías sobre la capacidad de reconocimiento visual de las diferentes máquinas.

En 1972 aparece el lenguaje PROLOG basado en las teorías de Minsky.

En 1973 se desarrolla el sistema experto llamado TEIRESIAS. El cometido de este sistema experto era el de servir de intérprete entre MYCIN y los especialistas que lo manejaban, a la hora introducir nuevos conocimientos en su base de datos. El especialista debía utilizar MYCIN de una forma normal, y cuando este cometiera un error en un diagnóstico (hecho producido por la falta o fallo de información en el árbol de desarrollo de teorías) TEIRESIAS corregiría dicho fallo destruyendo la regla si es falsa o ampliándola si es eso lo que se necesita.

En 1979 aparece XCON, primer programa que sale del laboratorio Su usuario fue la Digital Equipment Corporation (DEC).

El cometido de XCON sería configurar todos los ordenadores que saliesen de la DEC. El proyecto presentó resultados positivos y se empezó a trabajar en el proyecto más en serio en diciembre de 1978.

En abril de 1979 el equipo de investigación que lo había diseñado pensó que ya estaba preparado para salir, y fue entonces, cuando se hizo una prueba real, esperando resolver positivamente un 95% de las configuraciones, este porcentaje tal alto se quedó en un 20% al ser contrastado con la realidad; XCON volvió al laboratorio, donde fue revisado y a finales de ese mismo año funcionó con resultados positivos en la DEC.

En 1980 se instauró totalmente en DEC. Y en 1984, el XCOM había crecido hasta multiplicarse por diez. El XCOM supuso un ahorro de cuarenta millones de dólares al año para la DEC.

Entre los años 80 a 85 se produce la revolución de los Sistemas Expertos

En estos 5 años se crearon diversos sistemas expertos como el DELTA, de General Electric Company, para la reparación de locomotoras diesel y eléctricas. "Aldo en Disco" para la reparación de calderas hidroestáticas giratorias usadas para la eliminación de bacterias.

Se crearon multitud de empresas dedicadas a los sistemas expertos como Teknowledge Inc., Carnegie Group, Symbolics, Lisp Machines Inc., Thinking Machines Corporation, Cognitive Systems Inc. formando una inversión total de 300 millones de dólares. Los productos más importantes que creaban estas nuevas compañías eran las "máquinas Lisp", que se trataba de unos ordenadores que ejecutaban programas LISP con la misma rapidez que en un ordenador central, y el otro producto fueron las "herramientas de desarrollo de sistemas expertos".

En 1987 XCON empieza a no ser rentable. Los técnicos de DEC tuvieron que actualizar XCOM rápidamente llegándose a gastar más de dos millones de dólares al año para mantenimiento y algo parecido ocurrió con el DELTA..También en 1987 aparecieron los microordenadores Apple y compatibles IBM con una potencia parecida a los LISP. El software se transfirió a máquinas convencionales utilizando el lenguaje "C" lo que acabó con el LISP.

A partir de los 90 y con el desarrollo de la informática, se produce un amplio desarrollo en el campo de la IA y los sistemas expertos, pudiéndose afirmar que estos se han convertido en una herramienta habitual en determinadas empresas en la actualidad.

La evolución histórica de los métodos utilizados en el desarrollo de los sistemas expertos también se ha producido a medida que se ha ido desarrollando la IA y los diferentes métodos que se han empleado para su resolución. El desarrollo de lenguajes como LISP y PROLOG condicionaron esa evolución, así como investigaciones en diversos campos relacionados. Los primeros sistemas expertos que se desarrollaron en los años 60 eran capaces de resolver solo problemas basados en situaciones determinadas ,mediante sistemas de reglas .Es a partir de los 70 cuando se empiezan a resolver problemas basados en situaciones inciertas, basados en medidas difusas al principio y en redes probabilísticas con posterioridad

### **Campos de aplicación de un Sistema Experto**

La aplicación de Sistemas Expertos será adecuada allí donde los expertos dispongan de conocimientos complejos en un área muy delimitada, donde no existan algoritmos ya establecidos ( o donde los existentes no puedan solucionar algunos problemas).

Otro campo de aplicación es allí donde encontremos teorías que resulten prácticamente imposibles de analizar todos los casos teóricamente imaginables mediante algoritmos y en un espacio de tiempo relativamente corto y razonable.

el experto encuentra a menudo una solución al problema gracias a las informaciones que posee sobre el problema y a su experiencia. Mientras esta solución sea susceptible de repetición y el planteamiento del problema esté claro, existe un razonamiento que puede ser reproducido por un Sistema Experto.

Debido a que la estructuración e implementación del conocimiento del experto requiere una gran cantidad de trabajo, sólo valdrá la pena realizar el esfuerzo de crear un Sistema Experto cuando un conocimiento sea válido durante un largo espacio de tiempo y vaya a ser utilizado por el mayor número de personas.

Los Sistemas Expertos ofrecen ayuda para:

Evitar fallos en labores rutinarias complejas

Ampliar de forma más rápida los conocimientos de los especialistas.

Diagnosticar los fallos con mayor rapidez y conseguir tareas de planificación más completas y consistentes.

En este sentido, el Sistema Experto supone una descarga del experto en el trabajo rutinario y, por lo tanto, la reducción de sus problemas. Cuando la labor del experto no está tan sobrecargada, se reducen las decisiones erróneas y se aceleran los procesos en la toma de decisiones.

Los Sistemas Expertos no deben considerarse como soluciones aisladas respecto a otros desarrollos de software. La aplicación del software convencional debe realizarse allí donde tenga sentido hacerlo.

Otro punto que afecta al sistema es el hardware disponible. En una situación en la que un sistema sea necesario en muchos puntos distintos, el Sistema Experto deberá poder funcionar, a ser posible, en el hardware allí disponible. En la actualidad hay aparatos de

Inteligencia Artificial muy avanzados, pero que por cuestiones de coste (como siempre) no pueden ser aplicados en todos los puestos de trabajo de los usuarios.

### **Lenguajes de programación de sistemas expertos**

Los dos lenguajes más importantes en el desarrollo de Sistemas Expertos, para que podamos hacernos una idea de como se desarrolla este tipo de software "inteligente".

El nombre LISP es la abreviatura de List-Processing, ya que el LISP fue desarrollado para el procesamiento de listas. La lista es la estructura más importante de LISP. El lenguaje LISP fue diseñado ya a finales de los años 50 por McCarthy. A lo largo de los últimos años se han desarrollado muchos dialectos, por ejemplo MACLISP, COMMONLISP, INTERLISP, ZETALISP, donde el COMMONLISP se está imponiendo cada vez más como estándar.

En LISP se dan los siguientes conceptos característicos:

- **Listas y Átomos:** La estructura más importante es la lista. Los átomos pueden subordinarse a cualidades.
- **La Función:** Cada función LISP y cada programa LISP tiene estructura de lista. Los programas no pueden distinguirse sintácticamente de los datos. LISP ofrece sus propias funciones básicas.
- **Forma de Trabajo:** LISP es un lenguaje funcional. Ofrece la posibilidad de realizar definiciones recursivas de funciones. La unión de procedimientos se realiza de forma dinámica, es decir en plena ejecución, y no como en otros lenguajes de programación. El sistema realiza automáticamente una gestión dinámica de memoria.

Como se ha descrito antes, la estructura más importante en LISP es la lista:

(A (B C) D) es una lista con tres elementos

A átomo

( B C ) lista de átomos B y C

## D átomo

También está permitida una lista vacía , "( )" ó " NIL " , que significa lo mismo.

Con esta estructura podemos configurar estructuras de cualquier complejidad, tan grandes como queramos.

Los átomos son números , cadenas de caracteres o símbolos. Un símbolo puede tener varios valores, al igual que una variable en otros lenguajes de programación, como por ejemplo un número, o también puede ser el nombre de una función, o incluso ambos. Además a un símbolo pueden subordinarse cualidades, que además del valor del símbolo, contienen información adicional. Estas cualidades también reciben el nombre de atributos.

Para los más interesados en el funcionamiento de este lenguaje, voy a explicar brevemente alguna característica de cómo está implementado LISP, y de cómo "funciona por dentro".

### **Componentes de un sistema LISP.**

Un componente importante de un sistema LISP es la gestión dinámica de la memoria. El sistema administrará el espacio en la memoria para las listas en constante modificación, sin que el usuario lo deba solicitar. Libera los espacios de memoria que ya no son necesarios y los pone a disposición de usos posteriores. La necesidad de este proceso se deriva de la estructura básica de LISP, las listas, que se modifican de forma dinámica e ilimitada.

Además un sistema LISP abarca bastante más que el solo intérprete del lenguaje LISP. Consta de algunos cómodos módulos que ofrecen ayuda en el desarrollo y control del progreso en programas, como son el Editor, el File-System y el Trace. Por supuesto estos módulos sólo están en versiones de LISP que contengan la conocida interfaz gráfica IDE típica de los modernos lenguajes visuales. (IDE = entorno de desarrollo integrado).

**PROLOG** es la abreviatura de PROgramación LOGica, con lo que hacemos mención a la procedencia del lenguaje: Es una realización de lógica de predicados, como lenguaje de programación.

En la actualidad, el PROLOG se aplica como lenguaje de desarrollo en aplicaciones de Inteligencia Artificial en diferentes proyectos de Europa. En los Estados Unidos, el LISP

está más extendido que el PROLOG. Pero para la mayoría de los terminales de trabajo de Inteligencia Artificial se ofrece también el PROLOG.

Como una especie de semiestándar se han establecido el DECsystem-10 PROLOG de Edimburgo y el PROLOG descrito en el libro "PROGRAMMING IN PROLOG" de W.F.Clocksinn y C.S.Melish. La mayoría de los dialectos PROLOG se basan en este y contienen el DECsystem-10 PROLOG en su ámbito lingüístico.

Al contrario que el LISP (y otros lenguajes), en el PROLOG los programas son confeccionados de forma distinta. A los interesados en pueden leer a Kowalski que escribió un artículo con el título "Algorithm = Logic + Control". Con esto pretende decirnos que los algoritmos pueden ser divididos en su lógica y en sus mecanismos de control.

La lógica se representa en forma de predicados. Estos predicados aparecen en tres formas distintas: como hechos, como reglas y como preguntas. La lógica formulada como hechos y reglas se define como base de conocimientos. A esta base de conocimientos se le pueden formular preguntas.

Los mecanismos importantes del PROLOG son : recursividad, instanciación, verificación, unificación, backtracking e inversión.

- La **Recursividad** representa la estructura más importante en el desarrollo del programa. En la sintaxis del PROLOG no existen los bucles FOR ni los saltos; los bucles WHILE son de difícil incorporación, ya que las variables sólo pueden unificarse una sólo vez. La recursión es más apropiada que otras estructuras de desarrollo para procesar estructuras de datos recursivas como son las listas y destacan en estos casos por una representación más sencilla y de mayor claridad.
- La **Instanciación** es la unión de una variable a una constante o estructura. La variable ligada se comporta luego como una constante.
- La **Verificación** es el intento de derivar la estructura a comprobar de una pregunta desde la base de conocimientos, es decir, desde los hechos y reglas. Si es posible, la estructura es verdadera, en caso contrario es falsa.
- La **Unificación** es el componente principal de la verificación de estructuras. Una estructura estará comprobada cuando puede ser unificada con un hecho, o cuando

puede unificarse con la cabecera de una regla y las estructuras del cuerpo de dicha regla pueden ser verificadas.

## **Representación de Conocimiento**

Representación de conocimiento = Escribir en un lenguaje descripciones del mundo.

Representación - ``... un conjunto de convenciones sintácticas y semánticas que hacen posible el describir cosas' [Winston 74]

En IA, son descripciones para que una máquina inteligente las utilice.

**Sintaxis:** símbolos y conjunto de reglas para combinarlos.

**Semántica:** significado de las expresiones construidas.

A pesar de que todo programa de IA tiene una parte de base de conocimiento, es una área de investigación abierta.

Ingredientes básicos:

- un lenguaje de representación
- capacidad de inferencias
- conocimiento del dominio

### *El poder está en el conocimiento*

Una de las ambiciones es poder representar ``sentido común' '

El fin no es tanto buscar una explicación de comportamiento cognitivo racional, sino el poder construir razonablemente sistemas inteligentes

En general una representación debe de tener:

1. Expresividad Adecuada
2. Eficiencia de Razonamiento

Criterios para juzgar una representación:

- Capacidad Lógica: Que sea capaz de expresar el conocimiento que deseamos expresar.

- Poderío Heurístico: Capacidad para resolver problemas (inferencia).
- Conveniencia de la Notación: Simplicidad para acceder al conocimiento y facilidad de entendimiento (declarativa).

La representación determina la facilidad con la que podemos resolver ciertos problemas y utilizar el conocimiento [Marr 82].

Ejemplo: representación de números romanos vs arábigos (desarrollo del álgebra).

A nivel epistemológico:

- primitivos (cuáles y a qué nivel)
- meta-representaciones (permitir hacer razonamientos con el lenguaje)

Representaciones no cubiertas por lógica:

- definiciones vs hechos
- universales vs defaults
- razonamiento no-deductivo
- razonamiento no-monotónico

Representaciones alternas:

- procedurales (se necesita declarativo)
- analógico
- probabilístico

Problemas de representación de:

- sustancias (e.g., litro de leche)
- causalidad y tiempo
- creencias, deseos, intenciones, etc.

Consideraciones:

- Hacer explícito lo que se considere importante.
- Exhibir las restricciones inherentes al problema.
- Completo y preciso.
- Entendible.
- Fácil de usar.
- Computacionalmente factible.